Chapter 8 - Cryptography

Cryptography General Concepts

General Idea

Cryptography – the idea of storing and transmitting data in a form that only the authorized parties can interpret.

What services cryptosystems provide (669)

Cryptosystems provide the following services

- Confidentiality - secret
- Integrity – ensure things do not change
- Authentication – message comes from who you say it does
- Authorization – upon authentication, a user can be provided with a password to access a resource
- Non repudiation – ensure that no one can deny someone sent a message.

Definitions and Concepts (pg 670)

- Cryptography - a method of storing and transmitting data in a form only intended for authorized parties to read or process.
- Cryptanalysis* - science of studying, breaking, and reverse engineering algorithms and keys.
- Cryptology - the study of secret codes or ciphers and the devices used to create and decipher them (less specific than cryptanalysis, in face includes both terms above)

(more)

Cryptography definitions (670)

- Cryptosystem – A system or product that provides encryption and decryption
- Encryption – the method of transforming data (plaintext) into an unreadable format.
- Plaintext – the format (usually readable) of data before being encrypted
- Cipher text – the “Scrambled” format of data after being encrypted

(more)

Cryptography Definitions (670)

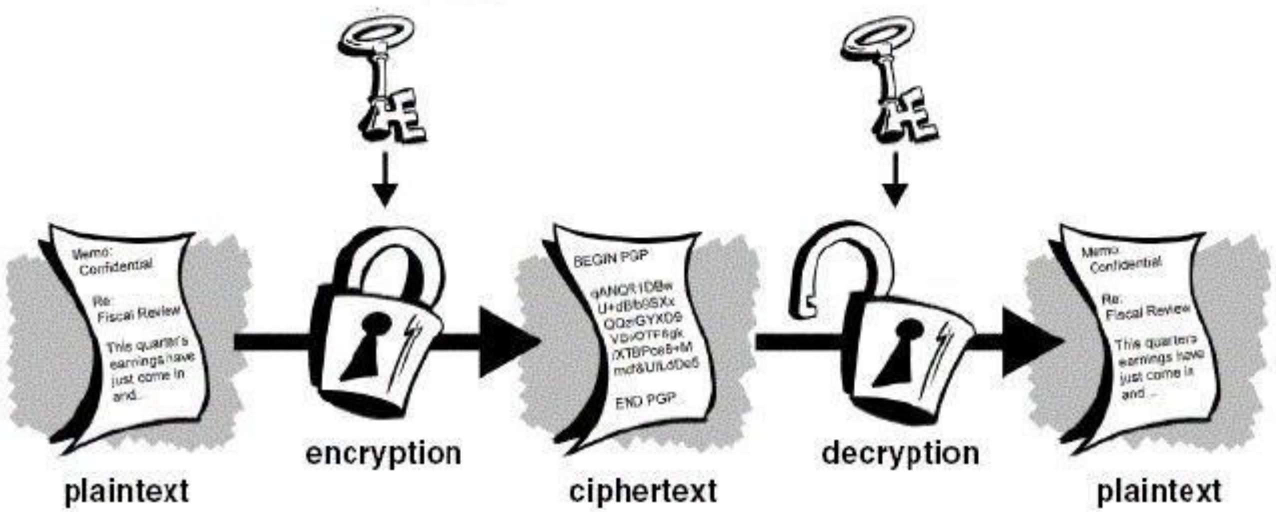
- Decryption – the method of turning cipher text back into
- Encryption algorithm – a set or rules or procedures that dictates how to encrypt and decrypt data.
- Key – (crypto variable) a values used in the encryption process to encrypt and decrypt

(more)

Cryptosystem Definitions (670)

- Key space – the range of possible values used to construct keys
- Key Clustering – Instance when two different keys generate the same cipher text from the same plaintext
- Work factor – estimated time and resources to break a cryptosystem

Basic Process (665)



Types of Encryption Ciphers (677)

Substitution

– Replaces one letter with another

Transposition

- Move letters around

Non Encryption Ciphers (673)

- Running Cipher – doesn’t use encryption, example. Find a certain book, turn to a certain page, then pick the letter from word 50 character 5.. An on and on to build a message.
- Concealment Cipher – a message within a message. Similar to running cipher but delivered in a single message.

(more)

Non Encryption Ciphers (674)

Stenography - The act of hiding data in plain site* (in another form). Such that nobody knows the secret data is there.

Does NOT encrypt data.

Example: Gif image, every 100 pixels are altered such they represent a number. This number is a value to be combined with every other 10 pixel values to be a message. (Your eyes wouldn’t detect the change in pixels)

Basic Tenants of Cryptography

BToC Confusion/Diffusion (685)

Strong Ciphers have the following attributes

- Confusion – commonly carried out through substitution
- Diffusion – commonly carried out through transposition (mixing up characters in message)

BToC (668)

Kerckhoffs Principal (668) - The only secrecy involved with a cryptosystem should be the key.*

This is important, let’s discuss, anyone have ideas why?

What is “security through obscurity?” anyone? Is a valid way to achieve security?

BToC

The goal of designing an encryption method is to make compromising it too expensive to be worth it*.

The amount of work to break it is called “work-factor”*

Protecting the key is important. There is no point to designing an encryption system that would take 1,000,000 years to break if you can easily just get some ones key!

Key Protection is CRITICAL*

(more)

BToC (key management) (734)

- Key lengths should be long enough to provide the necessary level of protection
- Keys should be stored and transported in a secure means (why?)
- Keys should be extremely random and use the full spectrum of the key space (why?)
- Keys lifetime should correspond with the sensitivity of the data to be protected

(more)

BToC Key Management (732)

- The more the key is used the shorter it’s lifetime should be
- Keys should be backed in case of emergency
- Keys should be destroyed when their lifetime is at and end.

Cryptography history (661)

Historical encryption algorithms

- Caesar cipher – just shift a few characters (A->E, B->F)
- Scytal Cipher – wrap a piece of tape around a certain sized cylinder such that the letters align to have the phrase. You must have the correct sized cylinder for the message to make sense. (661)
- Vignere Table – 2x2 matrix used for substitution (pg 663)

Methods of Encryption

Methods of Encryption Overview

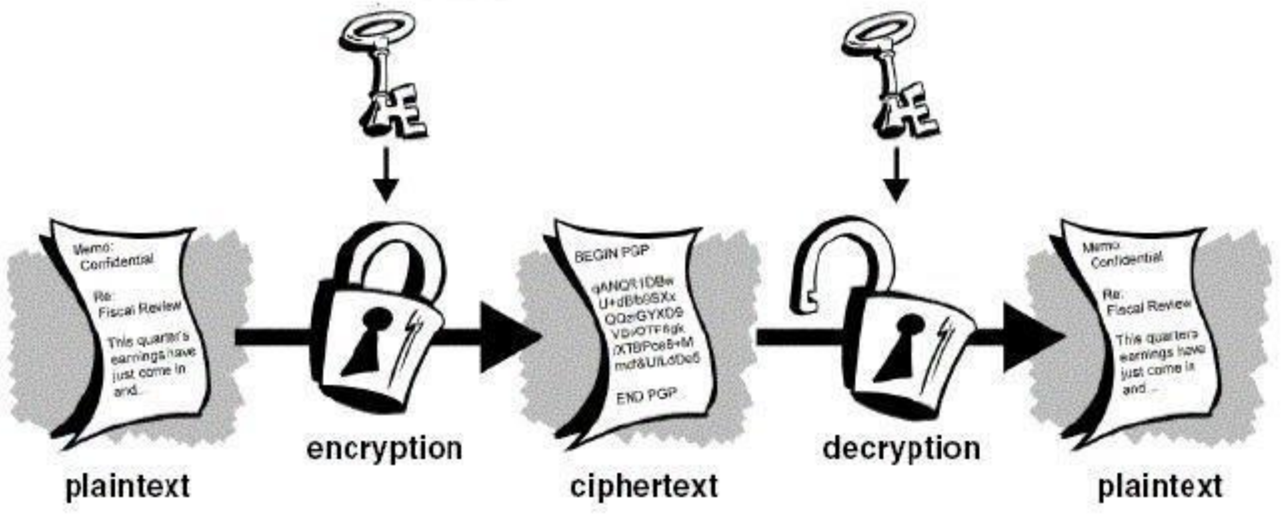
There are multiple “methods” of encryption

- Symmetric
- Asymmetric
- Hybrids
- Hashes (not really encryption, but no better place to put this) we are going to talk in detail about each of these

Symmetric Encryption

Symmetric Encryption (679)

Idea same key is used to BOTH encrypt and decrypt data!



Symmetric Pros (680)

- Fast
- Hard to break if using a large key size
- Provides Confidentiality

Symmetric Cons (681)

- Keys must be shared
  - This is difficult to really do? How to you get a key to someone you want to talk to?
  - Requires secure mechanism to deliver keys
  - Number of keys becomes needed becomes crazy large as number of people involved increases
  - Does Not provide Authenticity or Non-repudiation

Types of Symmetric Ciphers

- Block
- Stream

- Initialization Vectors

(more info on next pages)

Block (685)

- Break down message into fixed sized blocks, equal to the size of the key.
- Encrypt each block with the key.

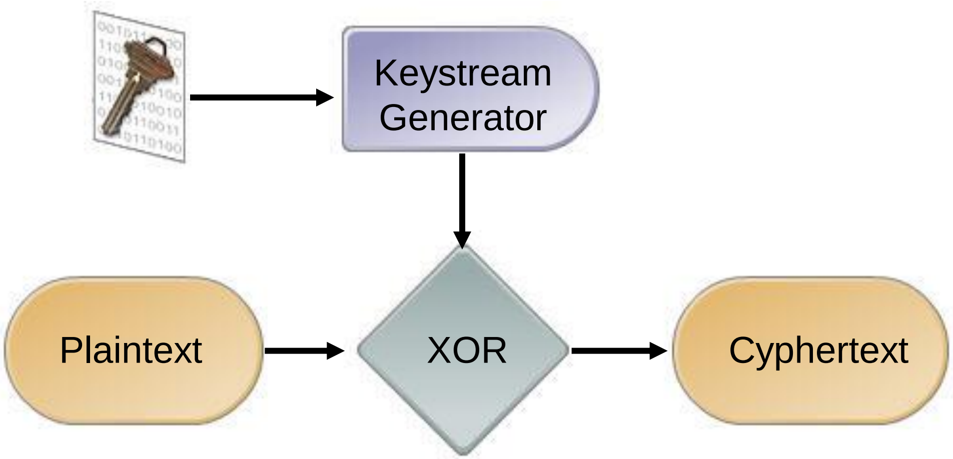
Create a diagram here

Stream (687)

- Do not break into blocks, instead take one character of the message at a time.
- The “key” is used with a “key stream generator” to create a stream of bits.
- These bits are XORed with the plaintext to create cipher text

(more)

Stream Cipher



Stream Cipher considerations

- Stream ciphers are hard work, better done in hardware*
- “key stream generator” should not generate repeating patterns.
- “key stream generator” should not product predictable output
- “key stream generator” should not produce a key stream related to the key
- The number of 0’s and 1s in the key stream should be about equal.

Initialization Vectors (688)

Not a type of symmetric encryption. They are random values that are used to ensure that patters are not created during the encryption process.

This is to allow you to generate different cipher text with the same plaintext and same key.

Used in both Stream and Block Ciphers

(Why would you want to do that?)

Now that we understand Symmetric Key concepts, let’s look at some SPECIFIC symmetric key cryptosystems.

Specific Symmetric Key Cryptosystems (algorithms)

Specific Symmetric Cryptosystems

- DES
- Triple DES
- AES
- IDEA
- Blowfish
- RC4
- RC5
- RC6

DES general info (696)

DES (Data Encryption Standard)

- Read history on 696 (on your own)
- Understand that DES is the “Standard” DEA is the actual algorithm.
- Retired when it became it was too easy to break.

DES (696)

- Symmetric algorithm
- Block based algorithm
- 64 bit key but really only 56 bits ?!?
- Divides the data into blocks and operates on them one at a time. These blocks are put through *16 rounds** (called an “S-box) of transposition (re-arranging) and substitution (changing) the order and type depends on the key.
- There are 5 “modes” of DES

DES Modes (overview)

- Electronic Code Book
- Cipher Block Chaining
- Cipher Feedback
- Output Feedback
- Counter Mode

ECB (698)

- Electronic Code Book – “regular” type of encryption, straight forward block by block encryption.
- Given the same plain text and the same key, the resulting cipher text will always be the same. (which Is bad as we’ll see later)

Cipher Block Chaining (700)

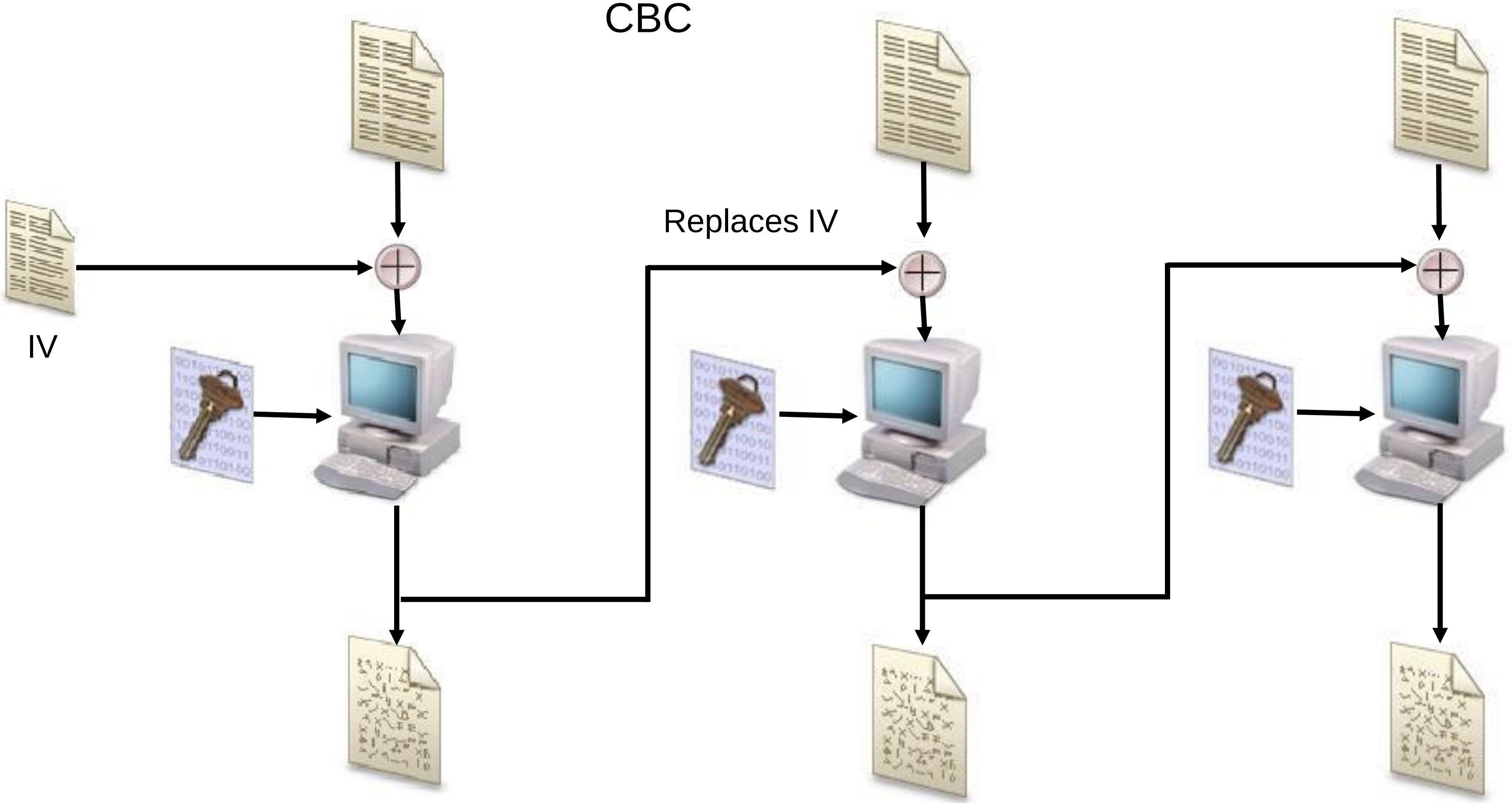
Tries to solve the problem of ECB mode.

- For each block of data to encrypt, CBC uses not only the key but the results from the previous block.
- For the first block (since we don’t have results from a previous block) we use an “Initialization Vector”

(see diagram on next page)

CBC diagram

CBC



Replaces IV

IV

Other DES Modes

Cipher Feedback, Output Feedback, Counter

I don’t think you’ll need to know much about these. I encourage you to read the sections in the book (700-702) yourself, and let me know if you have any questions.

Triple DES (703)

Like DES but uses *48 rounds** rather than DES's 16 rounds.

Has 4 rounds

- DES-EEE3 – 3 different keys: data is Encrypted, Encrypted, Encrypted (one key for each)
- DES-EDE3 – 3 keys: Encrypted, Decrypted, Encrypted
- DES- EEE2 – 2 keys, first and last operation use the same key
- DES- EDE2 – 2 keys, first and last operation use the same key

AES (703)

Advanced Encryption Standard – Developed to replace DES. There were multiple algorithms proposed to become “DES” the one chosen was called *Rijndael*.

- Block cipher
- Works well in software or hardware
- Low memory requirements
- Replaces DES
- Supports block sizes of 128, 192, 256 bits

IDEA (704)

International Data Encryption Algorithm

- Block cipher
- 128 bit key
- IDEA is faster than DES when implemented in software
- Used in PGP (later)
- patented

Blowfish (704)

- Block cipher
- 64 blocks of data
- Key length can be 32 – 448
- 16 rounds of cryptographic functions
- Unpatented, anyone can use it

RC4 (704)

- Owned by “RSA”
- Stream cipher
- Variable key size
- Used in SSL and in WEP (wireless) encryption
- Simple, fast and efficient
- Also called ARC4

RC5 (704)

- Owned by RSA
- Block cipher
- Block sizes of 32, 64, or 128 bits
- Key size can go up to 2048 bits
- “rounds” are not fixed, can be up to 255

RC6 (704)

- Same attributes at RC5, but modified to be faster
- Owned by RSA
- Block cipher
- Block sizes of 32, 64, or 128 bits
- Key size can go up to 2048 bits
- “rounds” are not fixed, can be up to 255
- faster than RC5

One Time Pad

One Time Pad

A modification of a symmetric key system.

- A “perfect cryptosystem”
- Unbreakable if implemented properly
- The key is a series of bits (0 and 1)
- The plain text is converted to bits
- The message is XORed with the pad/key to generated the cipher text (see next slide)

(more)

One Time Pad

1011 – plain text

0101 – pad

------ XOR

1110 – cipher text

- In a one time pad you use a different key/pad each time you send a message

One Time Pad considerations

- The pad must be used only one time
- The pad must be shared by both sides.
- The pad must be as long as the message
- The pad must be securely distributed
- The pad must be used up of truly random values

Asymmetric Encryption

Asymmetric Encryption (681)

Rather than use the same key for encryption and decryption, you use a different key for encryption and decryption

- These keys are mathematically related to each other

These keys are called

- Private Key
- Public

(more)

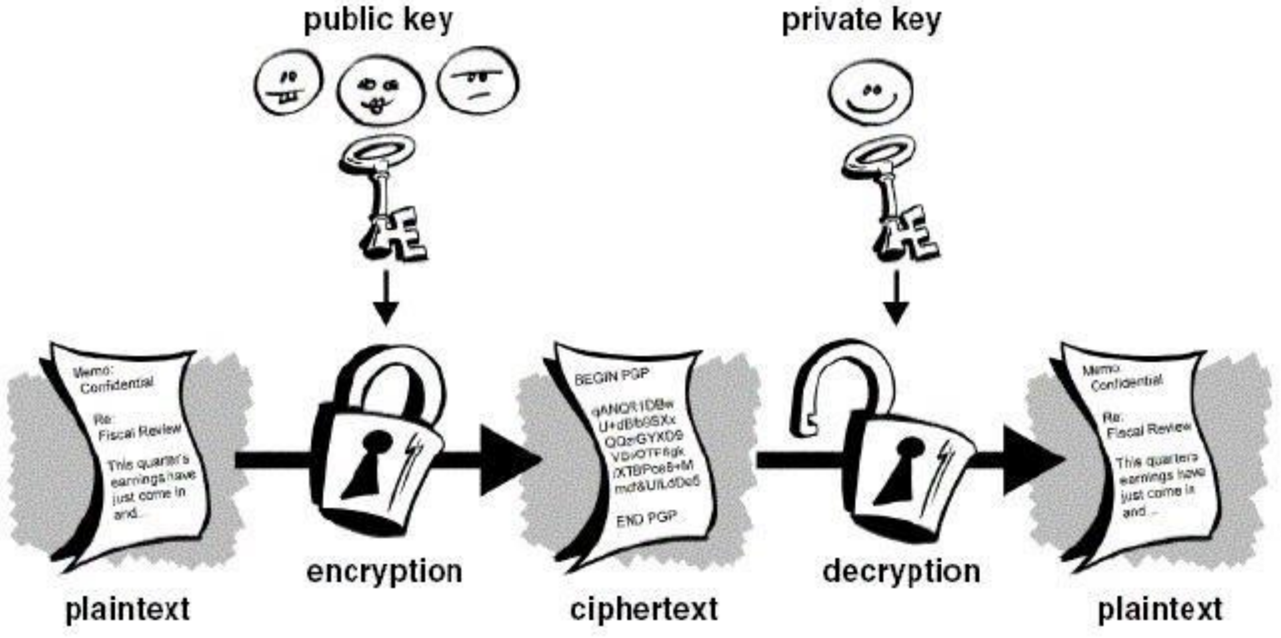
Asymmetric Encryption

Public Key – given to everyone

Private Key – stays secret

Asymmetric Encryption

Use public key to encrypt a message, private key can decrypt

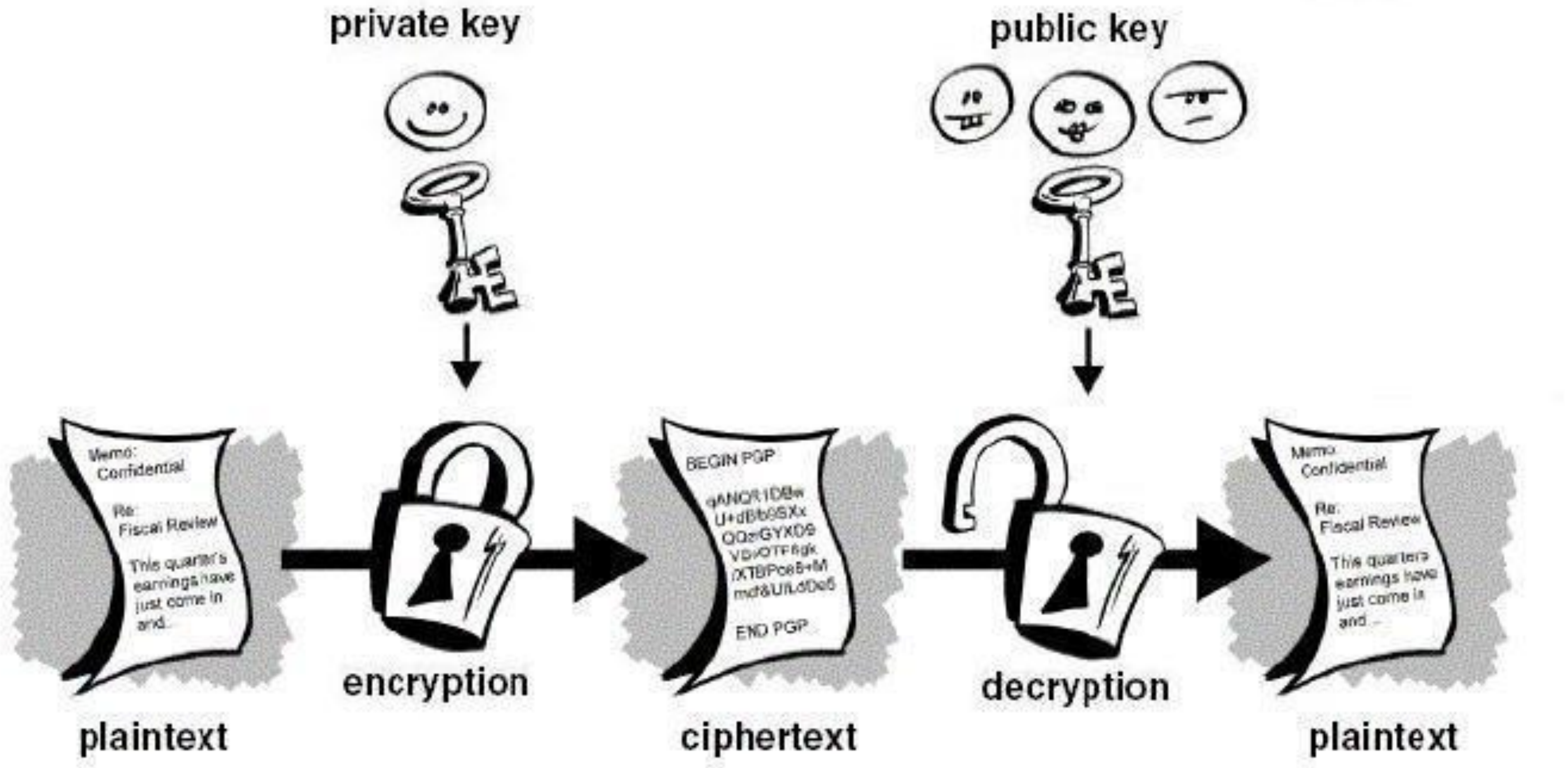


Asymmetric Encryption

Private and Public keys can actually do the reverse, you can use the private key to encrypt plaintext then the resultant cipher text can only be decrypted by the corresponding “public key”

(see diagram on next page)

Asymmetric Encryption (signing)



Signing

This process of using a private key to encrypt something that can only be decrypted with your public key is call “signing” and is used for authentication and non-repudiation

- If someone can read something you signed it proves that your private key was used.

One way function (710)

An important concept in symmetric encryption is a “One way function”

A one way function is an operation that is faster to complete in one direction than the other.

Example: if you drop a glass it breaks instantly to “undo” this would take much more time.

Asymmetric algorithms use this concept

(more)

One way functions (710)

- With Asymmetric encryption, a message is encoded with a one way function. This function supplies a trapdoor* (knowledge of how to undo the one way function faster). The private key can be used to retrieve this “trapdoor” and then use the trapdoor to put things back in order.
- Asymmetric algorithms use mathematical operations that are easier to do in one direction, than the other.

Asymmetric Pros/Cons (683)

Pros

- Key distribution is easy
- Scalable due to that
- Can provide authentication and non-repudiation

Cons

- Very mathematically intense
- Slow due to that

Specific Asymmetric Cryptosystems

Specific Asymmetric Cryptosystems

- Diffie-Hellman
- RSA
- El Gamal
- Elliptic Curve Cryptosystem

Diffie Hellman (706)

- Developed to address shortfalls of key distribution in symmetric key distribution.*
- Enables two people to receive a symmetric key securely without a previous relationship*
- Algorithm is based on “difficulty of calculating discrete logarithms in a finite field”* (I really don’t know what this means ;)
- Vulnerable to “man in the middle” attacks* (pg 707)

RSA (708)

- Can be used for digital signatures, key exchanges*, and encryption
- Security comes from the difficulty of factoring larges numbers.
- Private and Public keys are functions (results of mathematical operations) of large prime numbers.

El-Gamal (711)

- Used for digital signatures, encryption, and key exchange.
- based on calculating discrete logarithms in a finite field
- Actually an extension of Diffie-Hellman
- Slowest of all the asymmetric algorithms we will discuss.

Elliptic Curve Cryptosystem (712)

- Used for digital signatures, encryption and key distribution
- The fastest asymmetric algorithm that we discuss*
- Deals with discrete logarithms of elliptic curve*.
- Because it’s fast and easy used on devices with limited resources* (example: cell phones)

Hybrids

Hybrids (689)

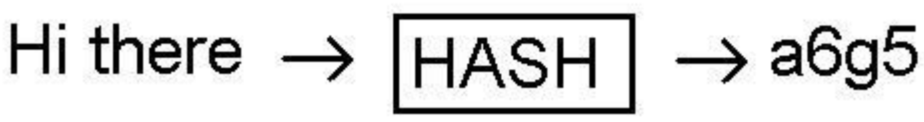
Hybrids cryptosystems use both Asymmetric and symmetric key cryptosystems.

- Use a Asymmetric system to encrypt a key to a symmetric key system. (i.e. to distribute the key).
- The Symmetric key is used to actually perform the encryption.
- This key is called a “session key”* and is only used for the current conversation.

Hashes

Hashes

A mathematical function that takes variable length input and produces a fixed length string.



Hash

- Since hashes take any length input and produce a fixed output, there will be multiple inputs that produce the same output, this is called a *collision**.
- A good hash function should not make it predictable on how to “force” a collision. Otherwise you could create a message what would generate the same hash as another (why is this bad?)

(more)

Hash

- Provide integrity, not confidentiality or authentication
- Hashes are vulnerable to man in the middle attacks (how)

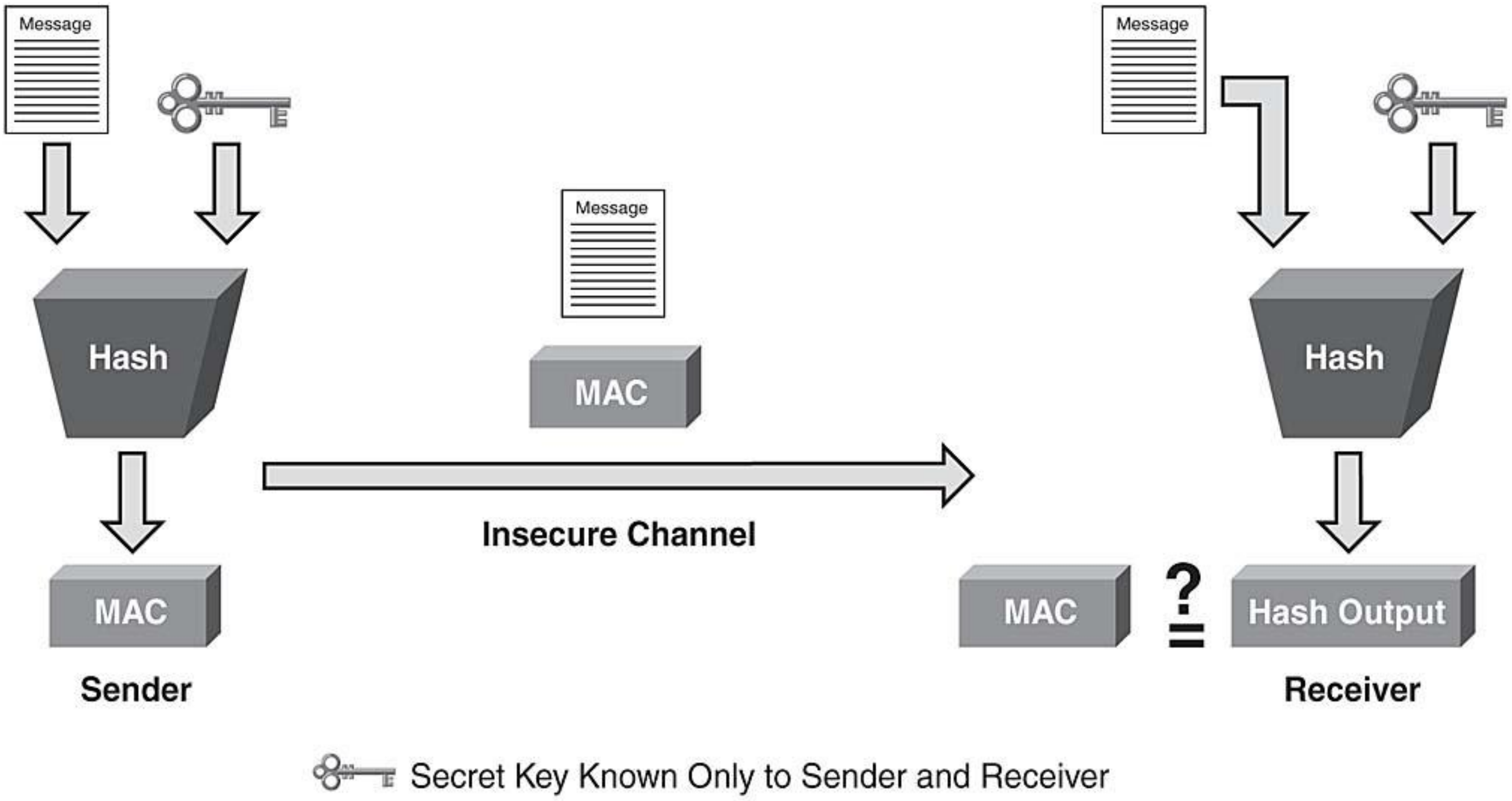
HMAC (715)

HMAC – uses a secret hey in combination to a hash algorithm to verify that a hash is not tampered with.

Rather than just doing the “hash algorithm” on the message, append your secret key to the message to create a new message and run the hash on the new message. The returned value is called a MAC (Message Authenticating Code)

(see diagram on next page)

HMAC (715)


Message
Message
Message
Hash
Hash
MAC
MAC
MAC
Insecure Channel
?
=
Hash Output
Sender
Receiver
Secret Key Known Only to Sender and Receiver

HMAC (715)

- Provide integrity and data original authentication (how?)
- Does not provide confidentiality
- Does not provide specific person authentication (as keys are shared)

CBC-MAC (717)

- Message is encrypted with a symmetric block cipher the final block of cipher text is used as the MAC.
- Sender sends the “plaintext” and the MAC.

CBC-MAC (717)

- Does not use a HASH
- Provides authentication and integrity
- Does not provide confidentiality

Specific Hash algorithm

Specific Hash algorithms

- MD2
- MD4
- MD5
- SHA

MD2 (719)

- Creates a 128 bit hash value, slower than MD4 and MD5

MD4 (719)

- creates 128 bit hash value
- Faster than MD2

MD5 (719)

- Creates 128 bit hash value
- More complex than MD2 and MD4
- More secure, harder to determine how to force collisions for a specific message

SHA (720)

- Designed to be used with the Digital Signature Standard, (for use with digital signatures)
- Creates 160 bit hash values
- SHA = SHA-1
- Alternate versions
  - SHA-256 = 256 bit hash values
  - SHA-385 = 384 bit hash values
  - SHA-512 = 512 bit hash values

Attacks against Hashes (721)

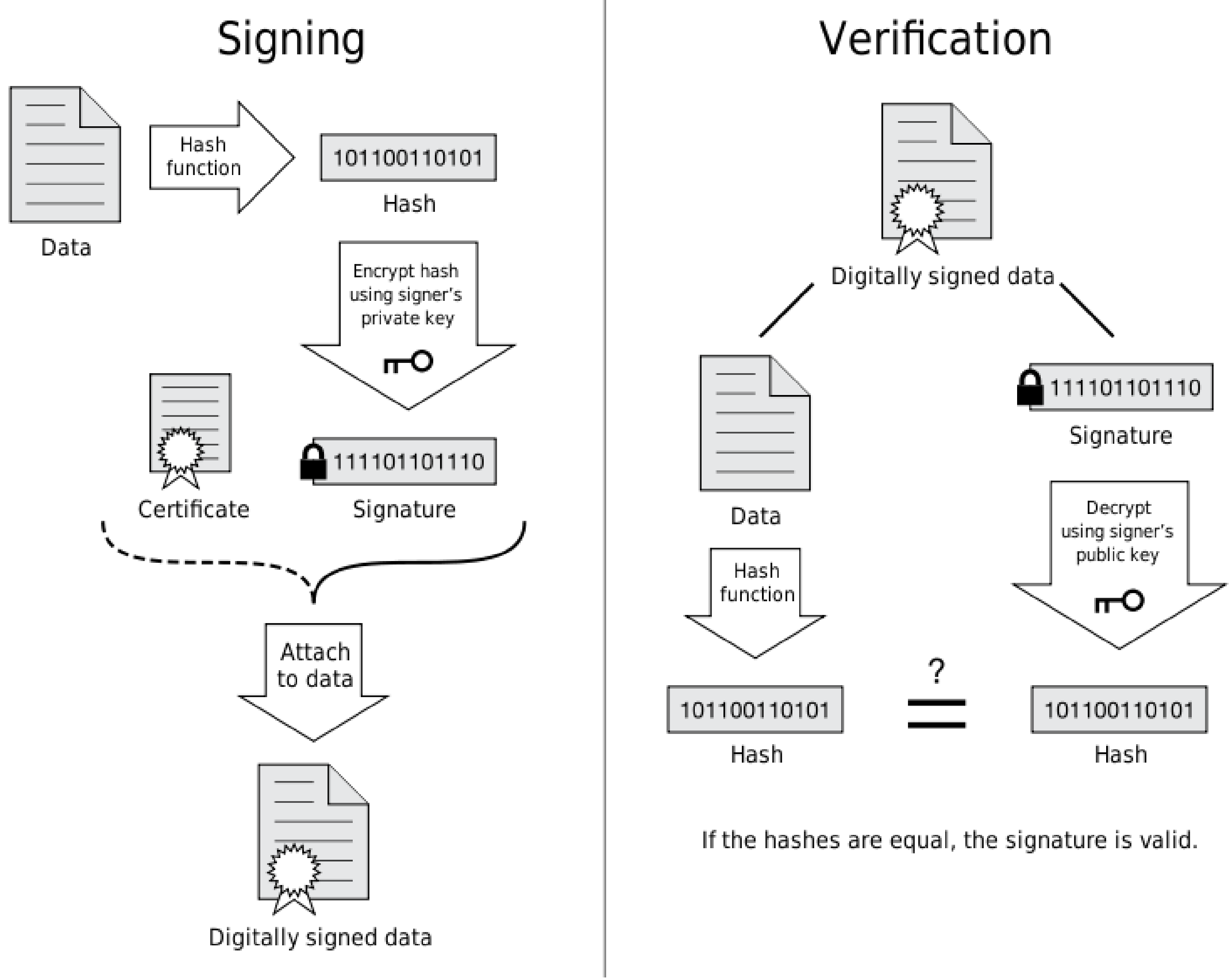
- Collisions – figure out how to create a message with the same hash value (collision)
  - Ex. “I’d like to buy 100 units of the widget” => A3BT
  - What if I could make the messages “I’d like to buy 500 units of the widget” and have the same hash value “A3BT” I can beat the integrity constraint
- This is called a birthday attack

OK… done with explaining the different types of cryptosystem! Let’s move on to how to apply them to do cool things!

Digital Signatures (722)

- We can use Asymmetric Cryptography and Hashes. To provide message authenticity, and Integrity and Non-repudiation.. Cool!

Digital Signature



If the hashes are equal, the signature is valid.

Digital Signatures

- How does this provide integrity?
- How does this provide non-repudiation?

PKI

PKI Generic Idea (725)

Public Key Infrastructure (PKI) is a series of programs, data formats, procedures, protocols, policies and public key (asymmetric) encryption. In order to provide secure communications for an organization.

Provides

- Authentication
- confidentiality
- No repudiation
- Integrity

PKI components (726)

- Each person has a digital “certificate*” which has information about a person, including a persons “public” key.
- The certificates are signed by a Certificate Authority*. By signing the Certificate the Certificate authority “vouches” for this persons certificate.

(more)

PKI components (729)

- A registration authority (RA) – establishes and confirms the identification of an individual. Once registered, the CA actually assignees, holds and distributes the Certificates.

PKI components (729)

- Certificate Authority signs certificates, and also provides a “Certificate Revocation List” (what’s a CRL?)

PKI steps (730)

1. User makes a request to RA
2. RA requests certain info from the user (like drivers license, address etc)
3. RA verifies user is who he says he is, and sends a request to create a cert to the CA.
4. CA creates a cert with users public key and identity information.

(more)

PKI steps (730)

5. Now when someone requests users info, the CA sends the certificate
6. The requesting user can extract the public key and knows that the information is valid as the CA also has signed the certificate.

PKI pros

- PKIs can provide the whole package of authentication, confidentiality, integrity and non-repudiation! This is awesome

PKI cons

- They are complex and hard to setup

Email Standards

Email Standards (737)

A great application of asymmetric and symmetric encryption is email!

MIME & SMIME (738)

- Email was designed to handle only text.
- MIME was created as a way to attach other types of data in email.
- MIME types specify what type of data is being attached.
- S/MIME is a secure version of MIME

PEM (738)

Privacy Enhanced Mail – Internet standard to provide secure email. Provides authentication, integrity, encryption and key management.

- AES for encryption (in CBC mode)
- RSA for authentication and key management
- X.509 certificates

Message Security Protocol

Military’s version of PEM

PGP (739)

Pretty Good Privacy

- Released as a freeware e-mail security program.
- First widespread use
- Uses IDEA for confidentiality
- Uses MD5 hash for integrity
- Certificates for identification and authentication
- Signed messages for non-repudiation
- Based on a “web of trust” where people verify each other identity..no strict structure